Deliverable 18

R&D STATUS REPORT

for

Contract DAAL01-97-C-0093

SBIR TOPIC A95-062

NETWORK SIMULATION OF TECHNICAL ARCHITECTURE

December 31, 1997

Submitted By:

PREDICTION SYSTEMS, INC.
309 Morris Avenue
Spring Lake, NJ 07762
☎ (908)449-6800
(908)449-0897
psi@predictsys.com

TABLE OF CONTENTS

1. TECHNICAL OBJECTIVES

The Technical Objectives provided here are taken from our proposal in response to SBIR Topic A95-062, Network Simulation of Technical Architecture. The General Simulation System (GSS), in conjunction with its supporting Model Development Graphics (MDG) and Run-Time Graphics (RTG) systems has been used by various government agencies over the past 10 years to simulate very large scale communications systems. There is now a multi-million dollar shelf of high quality, well documented GSS models, owned by the government, and in continuous active use running simulations to perform analyses and answer questions of current interest.

The objectives summarized below for raising RTG to a complete hierarchical run-time graphics facility will position the GSS family of simulation products and the existing shelf of GSS models at a unique level of ABCS simulation support readiness for many years into the future:

- Drawing icons using an RTG drawing board facility, including consideration for 3-D.
- Transforming icons from the drawing board into the RTG icon library with rotation and scaling.
- Transforming icons from the RTG icon library into a particular simulation as a normal or initial view, including rotation and scaling.
- Transforming icons from the RTG library into hierarchical iconic models for a particular simulation.
- Providing the modeler with generalized coordinate systems to be used in his specific simulations by his models.
- Performing transformations on hierarchical icons at various levels in the hierarchy, while viewing them at different levels in the hierarchy.
- Panning, rotating, and zooming on scenes (changing the viewpoint) in a simulation as hierarchical icons are undergoing their own transformation.
- Providing easy to use implementations of the above requirements so that the modeler can think in terms of his own problem and coordinate systems, with minimal effort to perform the transformations.
- Providing reasonable speed and accuracy of the operations in the implementation.

2. ICON LIBRARY MANAGEMENT (ILM) SYSTEM

The Icon Library Management (ILM) facility is a subsystem of the GSS Run-Time Graphics (RTG) system. The facility provides the ability to create and manage icons to be used in RTG. The facility includes extensive capabilities for icon images from standard graphics primitives. To support the development of icons, the ILM consists of 2 subsystems. The first subsystem is the ILM drawing board that provides the ability to create elementary icons. The other subsystem is the hierarchical icon builder that provides the ability to create icon hierarchies from elementary icons.

During December the ILM drawing board was updated in several areas. The most apparent difference is the change in the appearance of the drawing board by giving the buttons a 3-D appearance commonly found in most commercial applications. Figure 2-1 shows the new drawing board interface.

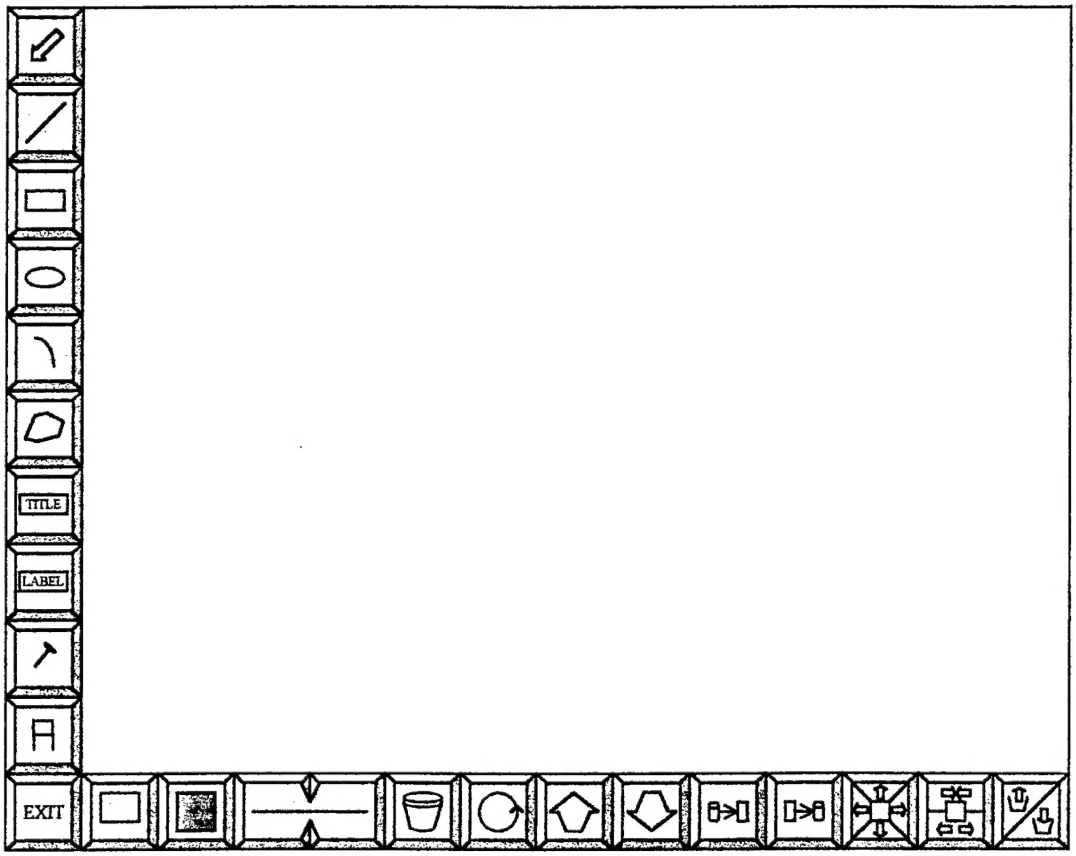

Figure 2-1. ILM Drawing Board

As seen in Figure 2-1, several new buttons have been added to the drawing board. These consist of the 3 buttons at the bottom right side of the screen and support the following functions:

- **Pan** – the pan button is used to pan the screen in the direction indicated by 1/3 the screen size. The button is divided into 5 fields. The four outer fields are used to pan in the up, down, right, and left directions. The middle field is used to reset the screen origin back to its original position (i.e., the origin).

- **Zoom** – The zoom button is used to quickly zoom in and out by a factor of 2. The button is divided into 3 fields. The top and bottom fields are used to zoom in and out respectively. The middle field is used to reset the zoom back to the original setting (i.e., unity).

- **Undo** – The undo button is used to remove the effect of recent operations. The button is divided into 2 fields. The top field performs the removal of the previous operation. Up to 4 operations can be undone. The bottom field reapplies the next operation (i.e., undo the undo).

Initial feedback from the ILM drawing board users, has resulted in several modifications to the ILM including the following:

- Ensure the Title and Label limits are not exceeded when copying selected items.

- Snap edges of the select region to the snap grid when rescaling.

- Store the icon name when read in for the default name used when saving an icon.

- Disallow infinitesimal primitives.

- Save icon and pin databases every time an icon is saved.

Initial work has also been performed in providing on-line help when using the ILM drawing board. This is performed by replacing the bottom row of option buttons with a text button that describes the features of the current operation being performed as shown in Figure 2-2. Once the operation is complete, the option buttons are reinstated.

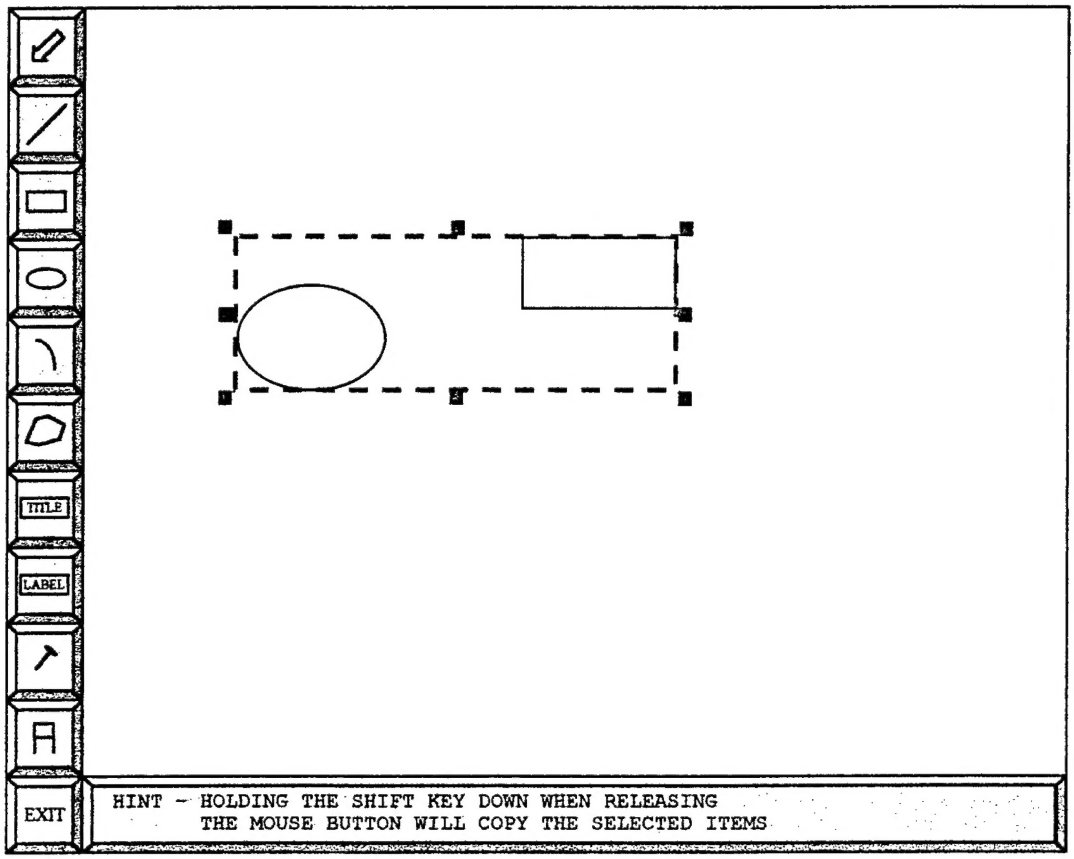

Figure 2-2. ILM Drawing Board Messages.

The ILM is continually evolving as each new capability is added. The current version of the ILM drawing board shown in Figure 2-1 supports the following capabilities:

- Handling mouse and keyboard events.
- Providing the drawing and option mode buttons.
- Creating new primitives including lines, rectangles, squares, ellipses, circles, arcs, polygons, titles, labels, pins, and text.
- Selecting primitives.
- Transformation operations on primitives including translation, copying, rotation, and scaling.
- Assigning and changing primitive attributes such as color, line style, and line thickness.

- Modifying the order that primitives are drawn using promote and demote capabilities.
- Storing and retrieving primitive information in a file.
- Selecting images using a graphical front-end interface.
- Supporting a snap mode.
- Displaying the snap grid.
- Use of hot keys.
- Option menu for changing ILM parameters.
- Pan and Zoom.
- X and Y axis.
- Text editing.
- Undoing operations.

3. RTG UPGRADE IMPLEMENTATION DESIGN DOCUMENT

A substantial review and update of the GSS-RTG User's Reference Manual was completed during December, and is being provided as Deliverable 24, RTG Upgrade Implementation Design Document. This version of the manual reflects a substantial amount of live testing of the new RTG system. The following summary provides highlights of some of the upgrades included in this version of the manual:

- *Line-state* and *color* attributes have been refined, and much of the general statement syntax has been updated for ease of use and consistency between functions.
- The handling of events coming into the simulation from RTG has been automated, so that the simulation will automatically invoke a designated GSS *event-handler* process whenever a reportable RTG event has occurred.
- The interrogation of incoming events as to type and lower level qualifiers has also been simplified with the revised RTG-GSS event template.
- User-defined menus have been modified to be more easily modeled, and to provide better functionality during simulation run time.
- The plot facility has been separated from the instrument symbol category, and is now supported by a plot resource.
- The symbol databases have been upgraded to reflect the latest data requirements supporting *icon*, *line*, *instrument*, and *pin* data.
- The method for querying and traversing multiple levels of icon hierarchy in the GSS-RTG shared databases from the GSS side has been simplified.

4. SIMULATION - GRAPHICS INTERFACE

Work on the operational interface between the GSS base and RTG systems continued throughout the December reporting period with revisions to the GSS editors and translators in support of the upgrades summarized in Section 3, above, and presented in detail in the upgraded RTG User's Reference Manual. Section 2 of the manual continues to provide a *build to* specification with respect to the functionality and format of the RTG subset of GSS statements. As shown in Figure 4-1, the GSS simulation and RTG tasks are running concurrently when graphics is active, and data is continually being shared via an intertask resource. As the simulation arrives at a graphic statement during the course of its execution, e.g., *insert icon*, critical information such as the identity and location of the icon is passed to the intertask resource on the GSS side, and GSS notifies RTG of an incoming event. RTG then acknowledges this event, receives the information from the intertask resource, and in the example above, displays the inserted icon on the graphic terminal.

The graphic statements described in Section 2 of the manual have been designed to be easy to understand and utilize from the user's perspective. This is made possible by burying all of the complexity behind the scenes, both in support of the preparation of GSS processes and resources, and in support of simulation execution. In the case of GSS process preparation, all graphic statements are edited for proper syntax and consistent use of attribute names from among the associated resources. When this editing process is complete, the statements are then translated such that the passing of pertinent information and notification of the RTG task will execute properly when invoked at simulation run time.

As reported previously, modifications to all of the GSS statement editors and all of the clauses within each statement have been completed. In addition, the Code Generation subsystem has now been completed to support translation of remaining line and instrument functions, as well as the insertion of hierarchical icons. The GSS-RTG interface has been augmented so that it will access the proper icon, line and pin database elements as built by the Hierarchical Icon Builder subsystem of the ILM and defined in the user's simulation control specification. The Simulation Control Specification translation subsystem underwent major design revisions during December in order to support the remaining Control Specification requirements such as icon scale, world space, NVS to BDS ratio, and to easily support additional GSS-RTG upgrades that will certainly arise as the systems further evolve.

With the completion of the above activities on the GSS Language subsystem, GSS Simulation Control Specification translation subsystem, and Hierarchical Icon Builder subsystem of the ILM, PSI is now performing integrated testing as follows:

- Enter the ILM drawing board subsystem and construct a set of primitive icons, including pins, positions for title and labels, etc.

- Using the primitive icons built in the drawing board, enter the ILM hierarchical icon builder subsystem and generate a set of hierarchical icons using elementary icons, pins, and lines across as many levels of icon hierarchy as desired

- Now enter the GSS base system and define a Simulation Control Specification which names icons built in the ILM, and relates them to attributes defined as type *icon* in the model resource descriptions

- Run the above simulation and exercise the various interactive RTG activities, as supported by either the RTG menu subsystem or directly on the scene. Also exercise the various GSS language statements comprising the RTG graphic subset.

Testing at this level provides the dual benefit of assessing GSS-RTG at a reasonably high level as relates to the modeler's overall functional objective, while continuing to validate the GSS statement edit and code generation facilities.

5. SUMMARY AND PLANS

Key activities performed on the ILM system during December include updating the drawing board interface and providing a help message facility. In addition, the interface between the ILM, GSS and RTG systems has been upgraded with major revisions to the Simulation Control Specification translator as regards input of symbols built by the ILM into a graphic simulation, and specification of graphic entities in the Graphic Section of the control specification. This subsystem defines the subset of icons in a user's ILM library, which are to be used in a particular simulation, and provides the operational link between the ILM and RTG. This subsystem has been upgraded to support the designation of an RTG_EVENT_HANDLER process within the simulation, which will be invoked automatically whenever an RTG event has occurred.

Formal deliverables are being provided to Mr. Son Nguyen at the Army Research Laboratory, Mr. Chandu Sheth, CECOM, and Mr. Bill Scott, JIEO. Deliverables 18 and 24, effective 31 December 97, are now provided, as follows:

- Deliverable 18 - Progress Report Nine (this report).
- Deliverable 21 – RTG Upgrade Implementation Design Document

During the next reporting period, PSI will continue to test of all the statement upgrades developed in December, and the automatic RTG_EVENT_HANDLER facility will be tested in conjunction with interrogation of the event template and corresponding updates to the user model parameters.

PSI will also deliver the Icon Library User Feedback & Upgrade Report containing all user feedback received to date on the ILM.